Escaping the Curse of Spatial Partitioning: Matchings With Low Crossing Numbers and Their Applications

Mónika Csikós @
Université Gustave Eiffel, LIGM, Equipe A3SI, ESIEE Paris, Cité Descartes 2 boulevard Blaise Pascal, 93162 Noisy-le-Grand Cedex, France.

Nabil H. Mustafa @
Université Gustave Eiffel, LIGM, Equipe A3SI, ESIEE Paris, Cité Descartes 2 boulevard Blaise Pascal, 93162 Noisy-le-Grand Cedex, France.

Abstract

Given a set system $(X, \mathcal{S})$, constructing a matching of X with low crossing number is a key tool in combinatorics and algorithms. In this paper we present a new sampling-based algorithm which is applicable to finite set systems. Let $n = |X|$, $m = |\mathcal{S}|$ and assume that X has a perfect matching M such that any set in $\mathcal{S}$ crosses at most $\kappa = \Theta(n^\gamma)$ edges of M. Then our algorithm computes a perfect matching of X with expected crossing number at most $\frac{8}{\gamma} \cdot \kappa$, in expected time $O\left(n^{2-\gamma} \ln^2 n + mn^{1-\gamma} \ln m\right)$.

As an immediate consequence, we get improved bounds for constructing low-crossing matchings for a slew of both abstract and geometric problems, including many basic geometric set systems (e.g., balls in $\mathbb{R}^d$). This further implies improved algorithms for many well-studied problems such as construction of ϵ-approximations. Our work is related to two earlier themes: the work of Varadarajan (STOC '10) / Chan *et al.* (SODA '12) that avoids spatial partitionings for constructing ϵ-nets, and of Chan (DCG '12) that gives an optimal algorithm for matchings with respect to hyperplanes in $\mathbb{R}^d$.

Another major advantage of our method is its simplicity. An implementation in `C++` is available on Github; it is approximately 200 lines of basic code without any non-trivial data-structure. Since the start of the study of matchings with low-crossing numbers with respect to half-spaces in the 1980s, this is the first implementation made possible for dimensions larger than 2.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures → Computational geometry

Keywords and phrases Matchings, crossing numbers, approximations

Funding The work of the authors has been supported by the grants ANR ADDS (ANR-19-CE48-0005) and ANR SAGA (JCJC-14-CE25-0016-01).

1 Introduction

Given a set system $(X, \mathcal{S})$, we say that a set $S \in \mathcal{S}$ *crosses* a pair $\{x, y\} \subseteq X$ iff $|S \cap \{x, y\}| = 1$. Define the *crossing number* of a perfect matching (resp. a spanning tree) G of X with respect to $\mathcal{S}$ as the maximum number of edges of G crossed by any $S \in \mathcal{S}$. The main focus of this paper is on constructing perfect matchings of X with low crossing numbers with respect to $\mathcal{S}$.

Matchings with low crossing numbers were originally introduced by Welzl [34, 35] for the special case where X is a set of points in $\mathbb{R}^d$ and $\mathcal{S}$ is induced on X by half-spaces. His result was then generalized by Chazelle and Welzl [10] to a broader class of set systems, which together with an improvement due to Haussler [21], gives the following general theorem.

▶ **Theorem A.** *Let $(X, \mathcal{S})$ be a set system with $n = |X|$, and dual shatter function*[1] $\pi^*_{\mathcal{S}}(k) = O(k^d)$.

[1] The *dual shatter function* $\pi^*_{\mathcal{S}}$ of $(X, \mathcal{S})$ is defined as follows. For any $k \leq |\mathcal{S}|$, $\pi^*_{\mathcal{S}}(k)$ is the maximum number of equivalence classes on X defined by a k-element subfamily $\mathcal{R} \subseteq \mathcal{S}$, where $x, y \in X$ are equivalent with respect to $\mathcal{R}$ if x belongs to the same sets of $\mathcal{R}$ as y.

Then there exists a perfect matching of X with crossing number $O\left(n^{1-1/d}\right)$.

In the last 30 years, matchings and spanning trees with low crossing numbers have become important structures in combinatorics and geometry, with many applications such as low-discrepancy colorings, near-optimal constructions of ε-approximations, and range searching, to name a few (see the texts [30, 24, 9, 20]).

Previous constructions. Let $(X, \mathcal{S})$ be a set system with $n = |X|$, $m = |\mathcal{S}|$, and let κ denote the smallest integer such that X has a perfect matching (resp. spanning tree) with crossing number κ with respect to $\mathcal{S}$. We review previous constructions in two separate settings.

Abstract set systems. The original method of Welzl [34, 35, 10] builds a perfect matching using the multiplicative weight update (MWU) method. Briefly, the algorithm maintains a weight function π on $\mathcal{S}$, with initial weights set to 1. It selects edges iteratively, always choosing an edge that is guaranteed to be crossed by sets of low total weight in π; it then updates π based on the chosen edge. The algorithmic bottleneck is in finding such an edge: for an abstract set system without additional structure, this takes $O(n^2 m)$ time for each of the $n/2$ iterations.
Another approach for the abstract case was proposed by Har-Peled [19] (see also [15]). His result implies that if $\kappa = \Theta(n^\gamma)$ for some $\gamma \in [1/\log n, 1]$, then a spanning tree of crossing number $O(\kappa/\gamma)$ can be found by solving an LP on $\binom{n}{2}$ variables and $m + n$ constraints. Combining this with an efficient approximate LP solver (e.g., [11]) leads to a randomized $\tilde{O}(mn^2)$ time algorithm. The approximation factor can be further improved using a general framework of rounding fractional solutions of minimax integer programs with matroid constraints. This method gives a randomized algorithm that constructs a spanning tree with expected crossing number at most $\kappa + O(\sqrt{\kappa \log m})$ in time $\tilde{O}(mn^4 + n^8)$ [12].

Geometric set systems. Now we turn to the case where X is a set of n points in $\mathbb{R}^d$ and $\mathcal{S}$ consists of subsets of X that are induced by geometric objects. In this setting, improved bounds are made possible using spatial partitioning. The current-best algorithms for geometric set systems induced by half-spaces recursively construct *simplicial partitions*[2] using cuttings [23], stored in a hierarchical structure called the partition tree, which then at its base level gives a matching with low crossing number. This approach is used in the breakthrough result of Chan [7] who gave an $O(n \log n)$ time algorithm to build partition trees with respect to half-spaces in $\mathbb{R}^d$, which then imply the same for computing matchings with crossing number $O(n^{1-1/d})$.

While the use of cuttings—and more generally, spatial partitioning—gives $o(n^2)$ running times, progress remains blocked in several ways:

a) Simplicial partitions only exist in certain geometric settings. Indeed, as shown by Alon *et al.*[5], they do not always exist in settings satisfying the requirements of Theorem A (e.g., the projective plane). Furthermore, spatial partitioning is not possible when dealing with abstract set systems such as those arising in graph theory or learning theory.
b) Optimal bounds for constructing simplicial partitions are only known for the case of half-spaces; this is one of the main problems left open by Chan [7]. Despite a series of research for semi-algebraic set systems (using linearization, cuttings, and more recently, polynomial partitioning [3]), the bounds are still sub-optimal for polynomials of degree > 2, with super-exponential dependence on the dimension.

[2] Given a set P of points, a family $\mathcal{S}$ of geometric sets in $\mathbb{R}^d$ and an integer t, the goal is to partition P into t roughly equal-sized sets such that the boundary of each object in $\mathcal{S}$ intersects the convex-hull of few sets of this partition.

c) There are large constants in the asymptotic notation depending on the dimension d both in the running time as well as the crossing number bounds, due to the use of cuttings (see [14]). For instance, in Chan's algorithm the constants are quite large—Theorem 3.2 [7] requires $\delta \leq \frac{1}{d^2}$, $b = 22$ (see [22]), which then implies that it constructs a spanning tree with a guaranteed crossing number no better than $12 \cdot 22 \cdot d^4 n^{1-1/d}$; this is at least $20000 \cdot n^{1-1/d}$ even for $d = 3$. Furthermore, the actual construction running time is at least $264 \cdot d^2 n \log n$, not counting the typically large constants in the several complex data structures that the algorithm needs (simplex range searching in $\mathbb{R}^d$ with dynamic insertion; see [22] for a discussion of its practical aspects in $\mathbb{R}^2$).

d) Practical implementation of spatial partitioning in $\mathbb{R}^d$, $d > 2$, even cuttings for hyperplanes, remains an open problem in geometric computing. Cuttings have been implemented in the planar case [18], which have then been used recently for computing ε-approximations w.r.t. half-spaces in $\mathbb{R}^2$ [22]. In particular, for $d > 2$, we know of no implementations for low-crossing matchings.

Recently there have been algorithms proposed for ε-nets and ε-approximations that avoid spatial partitioning [32, 8, 28, 27]. Our work can be considered another step along this theme.

2 Our Results

We state our main result assuming that we have access to a *membership Oracle of* $(X, \mathcal{S})$, which for a given element $x \in X$ and a set $S \in \mathcal{S}$ returns whether $x \in S$. Our main theoretical result is the following.

▶ **Theorem 1.** *Let $(X, \mathcal{S})$ be a set system, $n = |X|$, $m = |\mathcal{S}|$. Let $a > 0, b$ and $\gamma \in \left[\frac{1}{\log n}, 1\right]$ be constants such that any $Y \subseteq X$ has a perfect matching with crossing number at most $a|Y|^\gamma + b$ and $a|Y|^\gamma + b \geq 12 \ln(|Y| \cdot |\mathcal{S}|_Y|)$. Then* BUILDMATCHING$\big((X, \mathcal{S}), a, b, \gamma\big)$ *computes a perfect matching of X with expected crossing number at most $\left(\frac{8a}{\gamma}\right) n^\gamma + 4b \log n$, and with an expected $O\left(n^{2-\gamma} \ln^2 n + mn^{1-\gamma} \ln m\right)$ calls to the membership Oracle of $(X, \mathcal{S})$.*

Remarks:

- The algorithm BUILDMATCHING is presented in Section 3.
- In this paper, we mainly focus on constructing perfect matchings with low crossing numbers. However, our method can easily be modified to construct a spanning tree or a spanning path with the same guarantees up to a constant factor. In fact, Theorem 1 implies improved algorithmic bounds for problems where spanning paths with low crossing numbers are used in abstract settings, we present two examples in Section 6.

Now we give a list of consequences of Theorem 1, divided into three topics. All stated crossing number and running time bounds are in expectation.

1. **Low-crossing matchings.** Our results improve upon several previous constructions, see Table 1. For abstract set systems with dual shatter function $\pi^*_{\mathcal{S}}(k) = O(k^d)$, we improve the running time from $\tilde{O}(mn^2)$ to $\tilde{O}(mn^{1/d})$. Further, we provide the first sub-quadratic time construction for matchings with asymptotically-optimal crossing number with respect to balls. For set systems induced by semialgebraic sets in $\mathbb{R}^d$ (each set defined by at most s polynomial inequalities of degree at most Δ), we significantly improve the crossing number guarantee by removing the exponential dependence on d. However in contrast to the previous best algorithm for this setup [3], our running time depends on m.
   Importantly, our method does not use spatial partitioning, which makes it possible to handle abstract set-systems, and geometric set systems in $\mathbb{R}^d$ (not only in $\mathbb{R}^2$) without additonal complications. The precise guarantees for various set systems and their proofs are presented in Section 4.

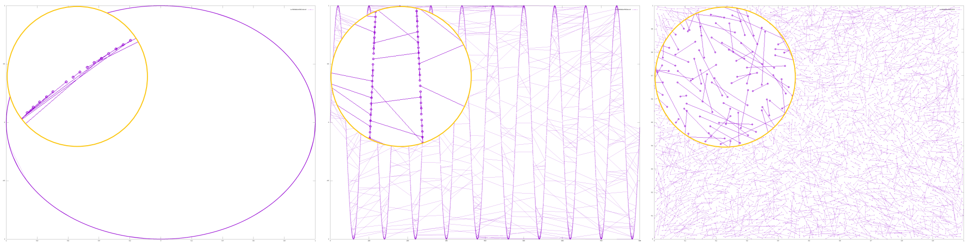
	Matchings / Spanning trees			
	Our method		Previous-best	
Set system	Crossing number	time	Crossing number	time
arbitrary with $\pi^*_{\mathcal{S}}(k) \leq ck^d$	$\left(\frac{8c^{1/d}d}{d-1} + o(1)\right) n^{1-1/d}$	$\tilde{O}\left(mn^{1/d}\right)$ (Corollary 9)	$O\left(n^{1-1/d}\right)$	$\tilde{O}(mn^2)$ [19, 11]
geometric induced by $\mathcal{H}_d$	$\left(6d^2 + o(d^2)\right) \cdot n^{1-1/d}$	$\tilde{O}\left(d^2 n^{1+1/d}\right)$ (Corollary 13)	$\geq 264 d^4 n^{1-1/d}$	$\tilde{O}(d^2 n)$ [7]
geometric induced by $\mathcal{B}_d$	$\left(6d^2 + o(d^2)\right) \cdot n^{1-1/d}$	$\tilde{O}\left(d^2 n^{1+2/d}\right)$ (Corollary 15)	$O\left(n^{1-1/d}\right)$	$O\left(n^{3+1/d}\right)$ [19, 11]
geometric induced by $\Gamma_{d,\Delta,s}$	$32e\Delta s n^{1-1/d} + o\left(n^{1-1/d}\right)$	$\tilde{O}\left(s\Delta^d mn^{1/d}\right)$ (Corollary 11)	$O\left(10^d s\Delta n^{1-1/d}\right)$	$O\left(n^{O(d^3)}\right)$ [3]

Table 1 Summary of our results for set systems $(X, \mathcal{S})$ with $n = |X|$, $m = |\mathcal{S}|$, and $n \leq m$. We use the notation $\pi^*_{\mathcal{S}}(\cdot)$ for the dual shatter function of $(X, \mathcal{S})$, $\mathcal{H}_d$ for half-spaces in $\mathbb{R}^d$, $\mathcal{B}_d$ for balls in $\mathbb{R}^d$, and $\Gamma_{d,\Delta,s}$ for semialgebraic ranges in $\mathbb{R}^d$ described by at most s equations of degree at most Δ (see Sec. 4).

2. Practical aspects. Our algorithm consists of $\frac{n}{2}$ iterations, where each iteration adjusts the weight of a random subset of $\binom{X}{2}$ and $\mathcal{S}$ and adds a randomly picked edge to the matching. The only black-box needed is the membership Oracle that returns for a given $x \in X$ and $S \in \mathcal{S}$, if $x \in S$. The time complexity of this operation depends on the precise way $(X, \mathcal{S})$ is given; typically this is independent of $|X|$ and $|\mathcal{S}|$ (using indexing, hashing).

A preliminary multi-threaded implementation in C++ for set systems induced on points by half-spaces in $\mathbb{R}^d$ is available on Github. It is approximately 200 lines of basic code without any non-trivial data-structures, being the first such implementation for $d > 2$.

The figures below show the matchings with respect to half-planes returned by our algorithm for $5,000$ points in $\mathbb{R}^2$ uniformly placed on a circle (in 17.39s), sine curve (in 17.17s), and randomly perturbed in a uniform grid (in 17.41s), each with a zoomed-in region. We find it surprising that our method, that is based only on random sampling, gives a matching that adapts so well to the each specific instance.



This makes progress towards the goals expressed at the end of the survey on range searching and its applications [1]: "*...an interesting open question is to develop simple data structures that work well under some assumptions on input points and query ranges*".

3. Discrepancy and approximations. By plugging in various upper-bounds on crossing numbers given by Theorem 1 and using techniques in Matoušek *et al.*[26], we immediately get improved construction bounds for discrepancy and ε-approximations. In particular, if d is a constant such that $(X, \mathcal{S})$ has dual shatter function $\pi^*_{\mathcal{S}}(k) = O(k^d)$, then we improve the running time of computing colorings with expected discrepancy $O\left(\sqrt{n^{1-1/d} \ln m}\right)$ from $O(mn^2)$ to $\tilde{O}(mn^{1/d})$. Moreover if in addition, $(X, \mathcal{S})$ has VC dimensionbounded by a constant D, then our method can be used to compute an ε-approximations of size $\tilde{O}\left(\left(\frac{D}{\varepsilon^2}\right)^{\frac{d}{d+1}}\right)$ in time $\tilde{O}\left(n+\left(D\left(\frac{D}{\varepsilon^2}\right)^{D+\frac{1}{d}}\right)\right)$, improving upon the previous-best time $O\left(n+\left(\frac{D}{\varepsilon^2}\right)^{D+2}\right)$. As these are standard applications of matchings with low crossing number, the proofs are omitted (see the survey [29]).

Organization. In Section 3, we describe our algorithm and prove Theorem 1. In Section 4, we show how Theorem 1 implies the bounds stated in Table 1 for various set systems. In Section 5, we present our experiments, and finally, in Section 6 we give some examples of applications in learning theory and graph theory.

3 Proof of Theorem 1

The proof rests on the following four key ideas:

1. We replace the bottleneck algorithmic step of finding a light edge in the multiplicative weights update technique by simply sampling an edge according to a carefully maintained distribution. In particular, we maintain weights not only on the sets in $\mathcal{S}$, but also on $\binom{X}{2}$. At each iteration we sample an edge e and a set S according to the current weights. Then we add e to our matching and update the weights by *doubling* the weight of each set that crosses e and *halving* the weight of each edge that is crossed by S. The idea of maintaining 'primal-dual' weights has been used earlier to approximately solve matrix games [17] and in geometric optimization [4].
2. In our case, the process is more elaborate as we are constructing a matching M at the same time as reweighing. Therefore, at the end of each iteration, as we add e to M, we are forced to set the weight of e and all edges adjacent to e to 0. This breaks down the reweighing scheme, as the removal of the edges amplifies the error introduced in later iterations and thus our maintained weights degrade over time. However, we prove that 'resetting' all the weights a logarithmic number of times suffices to ensure the required low crossing numbers.
3. The next idea is to show that an initial uniform random sample of $O\left(n \ln n\right)$ edges from $\binom{X}{2}$ already contains many good almost-matchings, and that it can be integrated in the proof to ensure that we end up, in expectation, with a good matching. We remark that this observation can be combined with some of the previous algorithms to improve their running times, though they remain $\Omega\left(mn\right)$.
4. This still does not get us to our goal as updating the weights of *all* edges and sets crossing the randomly picked set and edge would be too expensive. Instead, we show that updating the weights of a *uniform* sample of $O\left(n^{1-\gamma}\ln n\right)$ edges and $O(mn^{-\gamma}\ln m)$ sets at each iteration is sufficient for our purposes. The key observation here is that the standard multiplicative weights proof has an additive smaller-order term; we take advantage of this gap to improve the running time at the cost of amplifying this additive term, just enough so that it is still within a constant factor of the optimal solution.

Proof of Theorem 1. Later in this section, we prove the following statement for MATCHHALF.

▶ **Theorem 2.** *Let $(X,\mathcal{S})$ be a set system, $n=|X|$, $m=|\mathcal{S}|$, and let $\kappa > \max\{12\ln n, 2\ln m\}$ be such that any $Y \subseteq X$ has a perfect matching of crossing number at most κ with respect to $\mathcal{S}$. Let $E \subseteq \binom{X}{2}$ be a random edge-set obtained by adding each $e \in \binom{X}{2}$ to E independently with probability $12\ln(n)/n$. Then* MATCHHALF$\big((X,\mathcal{S}),E,\kappa\big)$ *returns a matching of size $n/4$ with expected crossing number at most 4κ, with an expected $O\left(n^2\ln^2(n)/\kappa + mn\ln(m)/\kappa\right)$ calls to the membership Oracle of $(X,\mathcal{S})$.*

Algorithm 1 $\textsc{BuildMatching}\big((X,\mathcal{S}),a,b,\gamma\big)$

$M \leftarrow \emptyset$
while $|X| \geq 4$ **do**
  $E \leftarrow \emptyset$
  for *each* $e \in \binom{X}{2}$ **do**
    add e to E with probability $12\ln(|X|)/|X|$
  $M' \leftarrow \textsc{MatchHalf}\left((X,\mathcal{S}),E,a|X|^{\gamma}+b\right)$ `//` M' `covers` $|X|/2$ `elements`
  $M \leftarrow M \cup M'$
  $X \leftarrow X \setminus \text{vertices}(M')$ `// remove elements covered by` M'
$M \leftarrow M \cup \{\text{edge connecting the remaining two elements of } X\}$
return M

The proof of Theorem 1 follows by applying Theorem 2 to each of the $\log n$ calls of MatchHalf. We get that the expected crossing number of the matching returned by $\textsc{BuildMatching}\big((X,\mathcal{S}),a,b,\gamma\big)$ is at most

$$4\sum_{i=0}^{\log n-1}\left[a\left(\frac{n}{2^i}\right)^{\gamma}+b\right] \leq \frac{8}{\gamma}\cdot an^{\gamma}+4b\log n,$$

and the overall expected number of calls to the membership Oracle is at most

$$\sum_{i=0}^{\log n-1} O\left(\frac{\left(\frac{n}{2^i}\right)^2\ln^2\left(\frac{n}{2^i}\right)}{a\left(\frac{n}{2^i}\right)^{\gamma}+b}+\frac{m\left(\frac{n}{2^i}\right)\ln m}{a\left(\frac{n}{2^i}\right)^{\gamma}+b}\right) = O\left(n^{2-\gamma}\ln^2 n + mn^{1-\gamma}\ln m\right).$$

◀

Proof of Theorem 2. The proof relies on the following technical lemma, whose proof is presented later in this section. For an edge e and a set S, we define $\mathrm{I}\left(e,S\right)$ to be 1 if S crosses e and 0 otherwise.

▶ **Lemma 3** (Main Lemma). *Let $\tilde{E}$ denote the set of edges that have non-zero weight when* $\textsc{MatchHalf}\big((X,\mathcal{S}),E,\kappa\big)$ *terminates. Then*

$$\mathbb{E}_{e_1,\ldots,e_{n/4}}\left[\max_{S\in\mathcal{S}}\sum_{i=1}^{n/4}\mathrm{I}\left(e_i,S\right)\right] \leq 2\cdot\mathbb{E}_{S_1,\ldots,S_{n/4}}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)\right]+\kappa. \tag{1}$$

The left-hand side of Equation (1) is precisely the expected crossing number of the edges returned by MatchHalf. The edge where the minimum in the right-hand side of Equation (1) is attained is commonly referred to as the 'shortest edge' with respect to $\{S_1,\ldots,S_{n/4}\}$. Note that we cannot use the classical short-edge lemma [10] directly in this setting as we need to find a short edge within a random set of edges $\tilde{E}$. Hence, we prove the following version of the short-edge lemma which is also sensitive to the crossing number κ.

▶ **Lemma 4.** *Let $(Y,\mathcal{R})$ be any set system and κ be such that Y has a perfect matching with crossing number κ with respect to $\mathcal{R}$. Then there are at least $|Y|/6$ edges spanned by the points of Y such that any of them is crossed by at most $\frac{3|\mathcal{R}|\kappa}{|Y|}$ sets of $\mathcal{R}$.*

Algorithm 2 $\textsc{MatchHalf}\big((X,\mathcal{S}),E,\kappa\big)$

$\omega_1(e) \leftarrow 1, \quad \pi_1(S) \leftarrow 1 \qquad \forall e \in E,\ S \in \mathcal{S}$
$\mathbf{p} \leftarrow 6\ln|E|/\kappa$
$\mathbf{q} \leftarrow 3\ln|\mathcal{S}|/\kappa$
for $i = 1,\ldots,n/4$ **do**
  $\omega_i(E) \leftarrow \sum_{e\in E}\omega_i(e)$
  $\pi_i(\mathcal{S}) \leftarrow \sum_{S\in\mathcal{S}}\pi_i(S)$
  choose $e_i \sim \omega_i$ // $\mathbb{P}[e_i = e] = \frac{\omega_i(e)}{\omega_i(E)} \quad \forall e \in E$
  choose $S_i \sim \pi_i$ // $\mathbb{P}[S_i = S] = \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \quad \forall S \in \mathcal{S}$
  $E_i \leftarrow$ sample from E with probability $\mathbf{p}$ // $\mathbb{P}[e \in E_i] = \mathbf{p} \quad \forall e \in E$
  $\mathcal{S}_i \leftarrow$ sample from $\mathcal{S}$ with probability $\mathbf{q}$ // $\mathbb{P}[S \in \mathcal{S}_i] = \mathbf{q} \quad \forall S \in \mathcal{S}$
  // $\mathrm{I}(e,S) = 1$ if e crosses S, $\mathrm{I}(e,S) = 0$ otherwise
  for $e \in E_i$ **do**
    $\omega_{i+1}(e) \leftarrow \omega_i(e)\big(1 - \frac{1}{2}\mathrm{I}(e,S_i)\big)$ // halve weight if S_i crosses e
  for $S \in \mathcal{S}_i$ **do**
    $\pi_{i+1}(S) \leftarrow \pi_i(S)\big(1 + \mathrm{I}(e_i,S)\big)$ // double weight if S crosses e_i
  set the weight in ω_{i+1} of e_i and of each edge adjacent to e_i to zero
return $\{e_1,\ldots,e_{n/4}\}$

Proof. Let M be a matching of Y such that any set of $\mathcal{R}$ crosses at most κ edges of M. Then there are at most $|\mathcal{R}|\cdot\kappa$ crossings between the edges of M and sets in $\mathcal{R}$. By the pigeonhole principle, there are at least $|M|/3 = |Y|/6$ edges in M such that each of them is crossed by at most

$$\frac{|\mathcal{R}|\cdot\kappa}{\frac{2}{3}\cdot|M|} = \frac{|\mathcal{R}|\cdot\kappa}{|Y|/3} = \frac{3|\mathcal{R}|\kappa}{|Y|}$$

sets of $\mathcal{R}$. ◀

Now we are ready to present the proof of Theorem 2, assuming the Main Lemma. First, we show that $\tilde{E}$ contains a short edge with high probability. Let $\tilde{X} \subset X$ denote the set of points that are not covered by the edges $\{e_1,\ldots,e_{n/4}\}$ returned by $\textsc{MatchHalf}\big((X,\mathcal{S}),E,\kappa\big)$. Applying Lemma 4 to $Y = \tilde{X}$ and $\mathcal{R} = \{S_1,\ldots,S_{n/4}\}$, we get that there is a set $E_{\text{short}} \subset \binom{\tilde{X}}{2}$ of at least $|\tilde{X}|/6 = n/12$ edges such that each $e \in E_{\text{short}}$ satisfies

$$\sum_{i=1}^{n/4}\mathrm{I}(e,S_i) \le \frac{3\cdot|\mathcal{R}|\cdot\kappa}{|\tilde{X}|} = \frac{3\cdot n/4\cdot\kappa}{n/2} = \frac{3}{2}\kappa. \tag{2}$$

We want to bound the probability of the event $\tilde{E}\cap E_{\text{short}} \neq \emptyset$. Observe that $\tilde{E} = E \cap \binom{\tilde{X}}{2}$ as we set the weight of an edge in E to zero if and only if it was equal or adjacent to some of $e_1,\ldots,e_{n/4}$. Thus $E\cap E_{\text{short}} = \tilde{E}\cap E_{\text{short}}$, and since each edge in E_{short} was added to E with probability $12\ln(n)/n$ independently, we get

$$\mathbb{P}\left[\,\tilde{E}\cap E_{\text{short}} \neq \emptyset\,\right] = \mathbb{P}\left[\,E\cap E_{\text{short}} \neq \emptyset\,\right] \ge 1 - \left(1 - \frac{\ln n}{n/12}\right)^{n/12} \ge 1 - \frac{1}{n}.$$

Since each edge in E_{short} crosses at most $\frac{3}{2}\kappa$ sets of $\{S_1, \dots, S_{n/4}\}$, we get

$$\mathbb{P}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)\leq\frac{3}{2}\kappa\right]\geq\mathbb{P}\left[\,E\cap E_{\text{short}}\neq\emptyset\,\right]\geq 1-\frac{1}{n}.$$

Now we return to Equation (1). We bound the expectation in the right-hand side using the fact that $\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)\leq\frac{n}{4}$ always holds

$$\mathbb{E}_{S_1,\dots,S_{n/4}}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)\right]$$
$$\leq\frac{3}{2}\kappa\cdot\mathbb{P}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)\leq\frac{3}{2}\kappa\right]+\frac{n}{4}\cdot\mathbb{P}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)>\frac{3}{2}\kappa\right]\leq\frac{3}{2}\kappa+\frac{n}{4}\cdot\frac{1}{n}.$$

Thus, by using the Main Lemma, the expected crossing number of the edges $\{e_1, \dots, e_{n/4}\}$ with respect to $\mathcal{S}$ can be bounded as

$$\mathbb{E}_{e_1,\dots,e_{n/4}}\left[\max_{S\in\mathcal{S}}\sum_{i=1}^{n/4}\mathrm{I}\left(e_i,S\right)\right]\leq 2\cdot\mathbb{E}_{S_1,\dots,S_{n/4}}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{n/4}\mathrm{I}\left(e,S_i\right)\right]+\kappa\leq 4\kappa.$$

Finally, we bound the number or membership Oracle calls. At each iteration $i = 1, \dots, n/4$, we update the weights of $|E_i| + |\mathcal{S}_i| = O\left(n\ln^2(n)/\kappa + m\ln(m)/\kappa\right)$ elements in expectation, each requiring one call to the membership Oracle. Thus in expectation, the total number of membership Oracle calls is $O\left(n^2\ln^2(n)/\kappa + mn\ln(m)/\kappa\right)$. This concludes the proof of Theorem 2. ◀

Proof of Main Lemma. The proof is subdivided into three lemmas. For brevity, we set $t = n/4$. The first lemma is proved by examining the total weight of the sets of $\mathcal{S}$ in π_{t+1}.

▶ **Lemma 5.**

$$\mathbb{E}_{e_1,\dots,e_t}\left[\max_{S\in\mathcal{S}}\sum_{i=1}^{t}\mathrm{I}\left(e_i,S\right)\right]\leq\frac{1}{\ln 2}\sum_{i=1}^{t}\mathbb{E}_{e_i}\left[\sum_{S\in\mathcal{S}}\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\mathrm{I}\left(e_i,S\right)\right]+\frac{\kappa}{3\ln 2}.$$

Proof. Let $\pi_{t+1}(\mathcal{S})$ denote the total weight of the sets of $\mathcal{S}$ in π_{t+1}. We bound $\pi_{t+1}(\mathcal{S})$ in two different ways. On the one hand, $\pi_{t+1}(\mathcal{S})$ is clearly lower-bounded by the weight of the set of maximum weight in π_{t+1}. Recall that the weight of a set S is doubled in iteration i if and only if S crosses e_i, therefore

$$\pi_{t+1}(\mathcal{S})\geq\max_{S\in\mathcal{S}}\pi_{t+1}(S)=2^{\max\limits_{S\in\mathcal{S}}\sum_{i=1}^{t}\mathrm{I}\left(e_i,S\right)\cdot\mathbb{1}_{\{S\in\mathcal{S}_i\}}},$$

where $\mathbb{1}_{\mathcal{A}}$ denotes the indicator whether an event $\mathcal{A}$ happens. On the other hand, we can express $\pi_{t+1}(\mathcal{S})$ using the update rule of the algorithm

$$\pi_{t+1}(\mathcal{S})=\sum_{S\in\mathcal{S}}\pi_{t+1}(S)=\sum_{S\in\mathcal{S}}\pi_t(S)\left(1+\mathrm{I}\left(e_t,S\right)\cdot\mathbb{1}_{\{S\in\mathcal{S}_t\}}\right)$$
$$=\sum_{S\in\mathcal{S}}\pi_t(S)+\sum_{S\in\mathcal{S}}\pi_t(S)\mathrm{I}\left(e_t,S\right)\cdot\mathbb{1}_{\{S\in\mathcal{S}_t\}}$$

$$= \pi_t(\mathcal{S}) + \pi_t(\mathcal{S}) \sum_{S \in \mathcal{S}} \frac{\pi_t(S)}{\pi_t(\mathcal{S})} \mathrm{I}(e_t, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_t\}}$$

$$= \pi_t(\mathcal{S}) \left(1 + \sum_{S \in \mathcal{S}} \frac{\pi_t(S)}{\pi_t(\mathcal{S})} \mathrm{I}(e_t, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_t\}}\right).$$

Unfolding this recursion and using the fact that $1 + a \le \exp(a)$, we get

$$\pi_{t+1}(\mathcal{S}) = \pi_1(\mathcal{S}) \prod_{i=1}^{t} \left(1 + \sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}}\right)$$

$$\le |\mathcal{S}| \cdot \exp\left(\sum_{i=1}^{t} \sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}}\right).$$

Putting together the obtained upper and lower bounds on $\pi_{t+1}(\mathcal{S})$, we get

$$2^{\max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}}} \le |\mathcal{S}| \cdot \exp\left(\sum_{i=1}^{t} \sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}}\right).$$

Taking the logarithm of each side yields

$$\ln(2) \cdot \max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}} \le \sum_{i=1}^{t} \sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}} + \ln |\mathcal{S}|. \quad (3)$$

Now we take the expectation with respect to the random edges $e_1, \ldots, e_t$ and the random collections of sets $\mathcal{S}_1, \ldots, \mathcal{S}_t$. Note that these edges and sets are picked independently.
First, we have a look at the left-hand side of Equation (3). Using linearity of expectation and the fact that $\mathbb{E}[\max\{X, Y\}] \ge \max\{\mathbb{E}[X], \mathbb{E}[Y]\}$ holds for random variables X and Y, we get

$$\mathbb{E}_{e_1,\ldots,e_t} \mathbb{E}_{\mathcal{S}_1,\ldots,\mathcal{S}_t} \left[\ln(2) \cdot \max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}} \right]$$

$$\ge \ln(2) \cdot \mathbb{E}_{e_1,\ldots,e_t} \left[\max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}(e_i, S) \cdot \mathbb{E}_{\mathcal{S}_i} \left[\mathbb{1}_{\{S \in \mathcal{S}_i\}} \right] \right]$$

$$= \ln(2) \cdot \mathbb{E}_{e_1,\ldots,e_t} \left[\max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}(e_i, S) \cdot \mathbb{P}\left[S \in \mathcal{S}_i \right] \right]$$

$$= \ln(2) \cdot \mathbf{q} \cdot \mathbb{E}_{e_1,\ldots,e_t} \left[\max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}(e_i, S) \right].$$

In the last step we used that $\mathbb{P}[S \in \mathcal{S}_i] = \mathbf{q}$ for all $S \in \mathcal{S}$. For the expectation of the right-hand side of Equation (3), we can write

$$\sum_{i=1}^{t} \mathbb{E}_{e_i} \mathbb{E}_{\mathcal{S}_i} \left[\sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{1}_{\{S \in \mathcal{S}_i\}} \right] + \ln |\mathcal{S}|$$

$$= \sum_{i=1}^{t} \mathbb{E}_{e_i} \left[\sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{E}_{\mathcal{S}_i} \left[\mathbb{1}_{\{S \in \mathcal{S}_i\}} \right] \right] + \ln |\mathcal{S}|$$

$$= \sum_{i=1}^{t} \mathbb{E}_{e_i} \left[\sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}(e_i, S) \cdot \mathbb{P}\left[S \in \mathcal{S}_i \right] \right] + \ln |\mathcal{S}|$$

$$= \mathbf{q} \cdot \sum_{i=1}^{t} \mathbb{E}_{e_i} \left[\sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}\left(e_i, S\right) \right] + \ln |\mathcal{S}| \,.$$

Hence Equation (3) implies

$$\ln(2) \cdot \mathbf{q} \cdot \mathbb{E}_{e_1,\ldots,e_t} \left[\max_{S \in \mathcal{S}} \sum_{i=1}^{t} \mathrm{I}\left(e_i, S\right) \right] \leq \mathbf{q} \cdot \sum_{i=1}^{t} \mathbb{E}_{e_i} \left[\sum_{S \in \mathcal{S}} \frac{\pi_i(S)}{\pi_i(\mathcal{S})} \mathrm{I}\left(e_i, S\right) \right] + \ln |\mathcal{S}|.$$

Dividing each side by $\ln(2) \cdot \mathbf{q} = (3 \ln 2 \ln |\mathcal{S}|)/\kappa$ gives the required inequality. ◀

The next lemma is proven by applying analogous arguments for the total weight of edges in ω_{t+1} with a small adjustment as in each iteration we set some edge weights to zero. Recall that $\tilde{E}$ denotes the set of edges that have non-zero weight in ω_{t+1}.

▶ **Lemma 6.**

$$\sum_{i=1}^{t} \mathbb{E}_{S_i} \left[\sum_{e \in E} \frac{\omega_i(e)}{\omega_i(E)} \mathrm{I}\left(e, S_i\right) \right] < 2\ln(2) \cdot \mathbb{E}_{S_1,\ldots,S_t} \left[\min_{e \in \tilde{E}} \sum_{i=1}^{t} \mathrm{I}\left(e, S_i\right) \right] + \frac{\kappa}{3}.$$

Proof. Let $\omega_{t+1}(E)$ denote the total weight of edges in ω_{t+1}. Again, we lower-bound $\omega_{t+1}(E)$ by the largest edge-weight in ω_{t+1}, which is now attained at some edge of $\tilde{E}$

$$\omega_{t+1}(E) \geq \max_{e \in E} \omega_{t+1}(e) = \max_{e \in \tilde{E}} \omega_{t+1}(e) = \left(\frac{1}{2}\right)^{\min_{e \in \tilde{E}} \sum_{i=1}^{t} \mathrm{I}\left(e, S_i\right) \cdot \mathbb{1}_{\{e \in E_i\}}}.$$

The upper bound is obtained by using the algorithm's weight update rule. Since e_t has positive weight in ω_t, but its weight in ω_{t+1} is set to 0, we have a strict inequality

$$\begin{aligned}
\omega_{t+1}(E) &= \sum_{e \in E} \omega_{t+1}(e) < \sum_{e \in E} \omega_t(e) \left(1 - \frac{1}{2} \mathrm{I}\left(e, S_t\right) \cdot \mathbb{1}_{\{e \in E_t\}}\right) \\
&= \sum_{e \in E} \omega_t(e) - \frac{1}{2} \sum_{e \in E} \omega_t(e) \mathrm{I}\left(e, S_t\right) \cdot \mathbb{1}_{\{e \in E_t\}} \\
&= \omega_t(E) \left(1 - \frac{1}{2} \sum_{e \in E} \frac{\omega_t(e)}{\omega_t(E)} \mathrm{I}\left(e, S_t\right) \cdot \mathbb{1}_{\{e \in E_t\}}\right).
\end{aligned}$$

Unfolding this recursion and using the fact that $1 + a \leq \exp(a)$, we get

$$\omega_{t+1}(E) \leq |E| \cdot \exp\left(-\frac{1}{2} \sum_{i=1}^{t} \sum_{e \in E} \frac{\omega_i(e)}{\omega_i(E)} \mathrm{I}\left(e, S_i\right) \cdot \mathbb{1}_{\{e \in E_i\}}\right).$$

Combining the obtained upper and the lower bounds on $\omega_{t+1}(E)$ and taking the logarithm of each side, we get

$$\ln\left(\frac{1}{2}\right) \cdot \min_{e \in \tilde{E}} \sum_{i=1}^{t} \mathrm{I}\left(e, S_i\right) \cdot \mathbb{1}_{\{e \in E_i\}} \leq -\frac{1}{2} \sum_{i=1}^{t} \sum_{e \in E} \frac{\omega_i(e)}{\omega_i(E)} \mathrm{I}\left(e, S_i\right) \cdot \mathbb{1}_{\{e \in E_i\}} + 2\ln |E|,$$

which is equivalent to

$$\sum_{i=1}^{t} \sum_{e \in E} \frac{\omega_i(e)}{\omega_i(E)} \mathrm{I}\left(e, S_i\right) \cdot \mathbb{1}_{\{e \in E_i\}} \leq 2\ln(2) \cdot \min_{e \in \tilde{E}} \sum_{i=1}^{t} \mathrm{I}\left(e, S_i\right) \cdot \mathbb{1}_{\{e \in E_i\}} + 2\ln |E|. \tag{4}$$

Now we take the expectation with respect to the random sets $S_1, \ldots, S_t$ and the random collections of edges $E_1, \ldots, E_t$. Note that these sets and edge-sets are picked independently. First look at the right-hand side of Equation (4). Using the linearity of expectation and the fact that $\mathbb{E}[\min\{X, Y\}] \leq \min\{\mathbb{E}[X], \mathbb{E}[Y]\}$ for random variables X and Y, we get

$$\mathbb{E}_{S_1,\ldots,S_t}\mathbb{E}_{E_1,\ldots,E_t}\left[2\ln(2)\cdot\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e, S_i\right)\cdot\mathbb{1}_{\{e\in E_i\}} + 2\ln|E|\right]$$

$$\leq 2\ln(2)\cdot\mathbb{E}_{S_1,\ldots,S_t}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e, S_i\right)\cdot\mathbb{E}_{E_i}\left[\,\mathbb{1}_{\{e\in E_i\}}\,\right]\right] + 2\ln|E|$$

$$= 2\ln(2)\cdot\mathbb{E}_{S_1,\ldots,S_t}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e, S_i\right)\cdot\mathbb{P}\left[\,e\in E_i\,\right]\right] + 2\ln|E|$$

$$= 2\ln(2)\cdot\mathbf{p}\cdot\mathbb{E}_{S_1,\ldots,S_t}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e, S_i\right)\right] + 2\ln|E|.$$

Again, the last equation follows as $\mathbb{P}\left[\,e\in E_i\,\right] = \mathbf{p}$ for all $e \in E$. Using the same argument as in the proof of Lemma 5, we can express the expectation of the left-hand side of Equation (4) as

$$\sum_{i=1}^{t}\mathbb{E}_{S_i}\mathbb{E}_{E_i}\left[\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e, S_i\right)\cdot\mathbb{1}_{\{e\in E_i\}}\right] = \mathbf{p}\cdot\sum_{i=1}^{t}\mathbb{E}_{S_i}\left[\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e, S_i\right)\right]$$

Thus Equation (4) implies

$$\mathbf{p}\cdot\sum_{i=1}^{t}\mathbb{E}_{S_i}\left[\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e, S_i\right)\right] \leq \mathbf{p}\cdot 2\ln(2)\cdot\mathbb{E}_{S_1,\ldots,S_t}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e, S_i\right)\right] + 2\ln|E|.$$

Dividing each side by $\mathbf{p} = (6\ln|E|)/\kappa$ gives the required inequality. ◀

We need one more lemma to tie the previous two together. The proof simply follows from the definition of expectation.

▶ **Lemma 7.** *For any $i \in [1, t]$, we have*

$$\mathbb{E}_{e_i}\left[\sum_{S\in\mathcal{S}}\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\mathrm{I}\left(e_i, S\right)\right] = \mathbb{E}_{S_i}\left[\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e, S_i\right)\right].$$

Proof.

$$\mathbb{E}_{e_i}\left[\sum_{S\in\mathcal{S}}\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\mathrm{I}\left(e_i, S\right)\right] = \sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\cdot\left(\sum_{S\in\mathcal{S}}\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\mathrm{I}\left(e, S\right)\right)$$

$$= \sum_{e\in E}\sum_{S\in\mathcal{S}}\frac{\omega_i(e)}{\omega_i(E)}\cdot\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\mathrm{I}\left(e, S\right)$$

$$= \sum_{S\in\mathcal{S}}\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\cdot\left(\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e, S\right)\right) = \mathbb{E}_{S_i}\left[\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e, S_i\right)\right].$$

◀

Finally, we combine Lemmas 5, 6, and 7 in the following way

$$\mathbb{E}_{e_1,\dots,e_t}\left[\max_{S\in\mathcal{S}}\sum_{i=1}^{t}\mathrm{I}\left(e_i,S\right)\right]$$

$$\le \frac{1}{\ln 2}\sum_{i=1}^{t}\mathbb{E}_{e_i}\left[\sum_{S\in\mathcal{S}}\frac{\pi_i(S)}{\pi_i(\mathcal{S})}\mathrm{I}\left(e_i,S\right)\right]+\frac{\kappa}{3\ln 2} \qquad \text{(Lemma 5)}$$

$$= \frac{1}{\ln 2}\sum_{i=1}^{t}\mathbb{E}_{S_i}\left[\sum_{e\in E}\frac{\omega_i(e)}{\omega_i(E)}\mathrm{I}\left(e,S_i\right)\right]+\frac{\kappa}{3\ln 2} \qquad \text{(Lemma 7)}$$

$$< \frac{1}{\ln 2}\left(\mathbb{E}_{S_1,\dots,S_t}\left[2\ln(2)\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e,S_i\right)\right]+\frac{\kappa}{3}\right)+\frac{\kappa}{3\ln 2} \qquad \text{(Lemma 6)}$$

$$\le 2\cdot\mathbb{E}_{S_1,\dots,S_t}\left[\min_{e\in\tilde{E}}\sum_{i=1}^{t}\mathrm{I}\left(e,S_i\right)\right]+\kappa.$$

This completes the proof of the Main Lemma and thus of Theorem 2. ◀

4 Corollaries of Theorem 1

Set systems with bounded dual shatter function. As before, let $(X,\mathcal{S})$ be a set system, $n=|X|$ and $m=|\mathcal{S}|$. We first recall the definition of the *dual shatter function* $\pi^*_{\mathcal{S}}$ of $(X,\mathcal{S})$. For any $\mathcal{R}\subseteq\mathcal{S}$, we say that the elements $x,y\in X$ are equivalent with respect to $\mathcal{R}$ if x belongs to the same sets of $\mathcal{R}$ as y. Then $\pi^*_{\mathcal{S}}(k)$ is defined as the maximum number of equivalence classes on X defined by a k-element subfamily $\mathcal{R}\subseteq\mathcal{S}$. The following theorem shows that set systems with polynomially bounded dual shatter function possess matchings with sublinear crossing number [24, Chap. 5.4].

▶ **Lemma 8.** *Let $(X,\mathcal{S})$ be a set system and c,d be constants such that $\pi^*_{\mathcal{S}}(k)\le ck^d$ for all $k\in[1,n]$. Then there is a perfect matching of X such that any set $S\in\mathcal{S}$ crosses at most $c^{1/d}n^{1-1/d}+\ln m$ edges of the matching.*

Observe that by definition, the dual shatter function of $(Y,\mathcal{S}|_Y)$ is upper-bounded by the dual shatter function of $(X,\mathcal{S})$ for any $Y\subseteq X$. Thus Lemma 8 implies that any $Y\subseteq X$ has a perfect matching with crossing number at most $c^{1/d}|Y|^{1-1/d}+\ln m$ with respect to $\mathcal{S}$. Applying Theorem 1 with we get the following corollary.

▶ **Corollary 9.** *Let $(X,\mathcal{S})$ be a set system and c,d be constants such that $\pi^*_{\mathcal{S}}(k)\le ck^d$ for all $k\in[1,n]$. Then* BuildMatching$\big((X,\mathcal{S}),\ c^{1/d},\ \ln m,\ 1-\frac{1}{d}\big)$ *returns a perfect matching of X with expected crossing number at most $\frac{8c^{1/d}d}{d-1}\cdot n^{1-1/d}+4\ln m\log n$ with an expected $\tilde{O}\left(n^{1+1/d}+mn^{1/d}\right)$ calls to the membership Oracle of $(X,\mathcal{S})$.*

Semialgebraic set systems. Let $\Gamma_{d,\Delta,s}$ denote all subsets of $\mathbb{R}^d$ such that each is induced by some semialgebraic set defined as the solution set of a Boolean combination of at most s polynomial inequalities of degree at most Δ. First, we give a bound on the dual shatter function of $\Gamma_{d,\Delta,s}$.

▶ **Lemma 10.** *Let $(X,\mathcal{S})$ be a set system such that X is a set of points in $\mathbb{R}^d$ and each set in $\mathcal{S}$ is induced by an element $\Gamma_{d,\Delta,s}$. Then the dual shatter function of $(X,\mathcal{S})$ can be upper-bounded as $\pi^*_{\mathcal{S}}(k)\le(4e\Delta s)^d\cdot k^d$.*

Proof. Let $\mathcal{R}\subseteq\Gamma_{d,\Delta,s}$ be a set of k ranges, defined by $\mathcal{P}=\{p_{ij}\ :\ 1\le i\le k, 1\le j\le s\}$, where each element is a d-variate polynomial of degree at most Δ. Observe that two points $x,y\in\mathbb{R}^d$ are equivalent with respect to $\mathcal{R}$ if sign $[\, p(x)\,] =$ sign $[\, p(y)\,]$ for all $p \in \mathcal{P}$. Therefore, $\pi^*_{\Gamma_{d,\Delta,s}}(k)$ can be upper-bounded by the number of different sign patterns in $\{-1,1\}^{ks}$ induced by ks d-variate polynomials of degree at most Δ. This quantity is bounded by $(4e\Delta s)^d \cdot k^d$, see [33, Theorem 3]. ◀

Now we can apply Corollary 9 and obtain the following guarantees for our algorithm.

▶ **Corollary 11.** *Let $(X, \mathcal{S})$ be a set system such that X is a set of n points in $\mathbb{R}^d$ and $\mathcal{S}$ consists of m subsets of X, each induced by an element of $\Gamma_{d,\Delta,s}$. Then* BUILDMATCHING$\big((X,\mathcal{S}),\ 4e\Delta s,\ \ln m,\ 1-\frac{1}{d}\big)$ *returns a perfect matching of X with expected crossing number at most $\frac{32e\Delta sd}{d-1} \cdot n^{1-1/d} + 4\ln m \log n$ with respect to $\mathcal{S}$ in expected time $\tilde{O}(s\Delta^d \cdot mn^{1/d})$.*

▶ Remark. The previous best algorithm for constructing matchings with low crossing numbers with respect to $\Gamma_{d,\Delta,s}$ relies on the polynomial partitioning technique [3]. It computes a perfect matching of n points in general position with crossing number $O(10^d s\Delta n^{1-1/d})$ with respect to *any* set in $\Gamma_{d,\Delta,s}$ in time $O(n^{O(d^3)})$, notably the running time is independent of m. Our algorithm provides improved running time bounds for specific instances with $m = n^{o(d^3)}$.

Half-spaces. Let $\mathcal{H}_d$ denote the set of all half-spaces in $\mathbb{R}^d$ and consider set systems on points in $\mathbb{R}^d$ induced by $\mathcal{H}_d$. For this setting, a typical pre-processing step is constructing a small-sized subfamily of $\mathcal{H}_d$—called a *test-set*—such that it suffices to construct a low-crossing matching with respect to this subfamily. We use a result of Matoušek [23] on test-sets, with a small addition: the bounds are stated with precise constants, in particular, with precise cutting constants from [14].

▶ **Lemma 12** (Test set lemma [23]). *Let X be a set of n points in $\mathbb{R}^d$, $\mathcal{H}_d$ be the set of all half-spaces in $\mathbb{R}^d$, and t be a parameter. There exists a set $\mathcal{T}(t)$ of at most $(d+1)t^d$ hyperplanes such that if a perfect matching of X has crossing number κ with respect to $\mathcal{T}(t)$, then its crossing number with respect to $\mathcal{H}_d$ is at most $(d+1)\kappa + \frac{6d^2n}{t}$.*

Now let X be a set of n points in $\mathbb{R}^d$ and $\mathcal{T} = \mathcal{T}(n^{1/d})$ be the set of $(d+1)n$ half-spaces in $\mathbb{R}^d$ provided by Lemma 12. Notice that $\mathcal{T} \subset \mathcal{H}_d = \Gamma_{d,1,1}$, thus by Lemma 10, $\pi^*_{\mathcal{T}}(k) \le (4e)^d k^d$. We apply Corollary 9 for $(X, \mathcal{T})$ and obtain the following.

▶ **Corollary 13.** *Let X be a set of n points in $\mathbb{R}^d$ and $\mathcal{T} = \mathcal{T}(n^{1/d})$ be the set of half-spaces provided by Lemma 12. Then* BUILDMATCHING$\big((X,\mathcal{T}),\ 4e,\ \ln n, 1-\frac{1}{d}\big)$ *returns a perfect matching of X with expected crossing number at most $\left[\, 6d^2 + (d+1)\cdot \frac{32ed}{d-1}\,\right] n^{1-1/d} + 4\ln^2 n$ with respect to half-spaces in $\mathbb{R}^d$, in expected time $O\left(d^2 n^{1+1/d} \ln^2 n\right)$.*

▶ Remark. The state-of-the-art algorithm for constructing matchings with low crossing number with respect to half-spaces is due to Chan [7]. While his method has a better dependence on n, there are large constants in the asymptotic notation: its crossing number guarantee is no better than $264d^4 n^{1-1/d}$ and the running time is at least $264d^2 n$. Moreover, its implementation is non-trivial and is only available in $\mathbb{R}^2$ [22].

Balls. Let $\mathcal{B}_d$ denote the subsets of X that are induced by balls in $\mathbb{R}^d$. It is well known that there are mappings $\alpha : X \to \mathbb{R}^{d+1}$ and $\beta : \mathcal{B}_d \to \mathcal{H}_{d+1}$ such that for any $x \in X$ and $B \in \mathcal{B}_d$, we have $x \in B$ iff $\alpha(x) \in \beta(B)$, see eg. [25, Chap. 10]. This mapping and Lemma 12 applied in $\mathbb{R}^{d+1}$ with $t = n^{1/d}$ give the following test set lemma for $\mathcal{B}_d$.

▶ **Lemma 14.** *Let X be a set of n points in $\mathbb{R}^d$. There exists a set $\mathcal{Q}$ of at most $(d+2)n^{1+1/d}$ balls such that if a perfect matching of X has crossing number κ with respect to $\mathcal{Q}$, then its crossing number with respect to $\mathcal{B}_d$ is at most $(d+2)\kappa + 6(d+1)^2 n^{1-1/d}$.*

Given a set X of n points in $\mathbb{R}^d$, let $\mathcal{Q}$ be the set of balls provided by Lemma 12. As $\mathcal{Q} \subset \mathcal{B}_d \subset \Gamma_{d,2,1}$, the dual shatter function of $\mathcal{Q}$ can be bounded as $\pi^*_{\mathcal{Q}}(k) \leq (8e)^d k^d$ (Lemma 10). We apply Corollary 9 for $(X, \mathcal{Q})$, and obtain the following corollary.

▶ **Corollary 15.** *Let X be a set of n points in $\mathbb{R}^d$ and let $\mathcal{Q}$ be the set of balls provided by Lemma 14. Then* BUILDMATCHING$\big((X, \mathcal{Q}),\ 8e,\ \ln(n^{1+1/d}),\ 1 - \frac{1}{d}\big)$ *returns a perfect matching of X with expected crossing number at most* $\left[\, 6(d+1)^2 + (d+2) \cdot \frac{64ed}{d-1} \,\right] n^{1-1/d} + \frac{4(d+1)}{d} \ln^2 n$ *with respect to balls in $\mathbb{R}^d$, in expected time $O(d^2 n^{1+2/d})$.*

▶ Remark. The previous-best algorithm to construct spanning trees with crossing number $O(n^{1-1/d})$ with respect to $\mathcal{B}_d$ is based on randomized LP rounding and has time complexity $\tilde{O}(mn^2)$ [19, 11]. Alternatively, one can obtain a matching with suboptimal crossing number $O(n^{1-1/(d+1)})$ by lifting X into $\mathbb{R}^{d+1}$, where the image of each range in $\mathcal{B}_d$ can be represented by a range in $\mathcal{H}_{d+1}$ and applying Chan's algorithm [7] with time complexity $O(n)$.

5 Empirical Aspects of BUILDMATCHING

In this section we present preliminary experimental results and provide some implementation details.

Experimental setup. We apply our algorithm for set systems induced by half-spaces in dimensions $2, 4, 6, 8$, and 10. We consider two different types of input point sets:

- **`Grid`**: each point is picked randomly in a cell of the uniform grid;
- **`Moment Curve`**: each point is a slightly perturbed element of the moment curve.

These examples capture two extremal cases: in the case of the **`Grid`** the optimal crossing number is $\Theta(n^{1-1/d})$, while it is $\Theta(d)$ in the case of the **`Moment Curve`** input. All the experiments are performed with dual Xeon E5-2643 v3 processors, each with 6 cores, 12 threads, at 3.4 GHz. We run our experiments with the parameters BUILDMATCHING$\big((X, \mathcal{T}),\ 0.6,\ 0,\ 1 - 1/d\big)$.

`Grid`

Input size	$d=2$ cr #	$d=2$ time (s)	$d=4$ cr #	$d=4$ time (s)	$d=6$ cr #	$d=6$ time (s)	$d=8$ cr #	$d=8$ time (s)	$d=10$ cr #	$d=10$ time (s)
10000	162	58.89	699	11.84	1238	8.07	1639	6.38	1863	6.73
25000	330	279.82	1509	37.33	2804	26.49	3912	20.32	4525	20.76
50000	630	918.26	2732	99.62	5251	61.21	7387	47.02	8797	48.66
100000	1170	3001.16	5040	271.29	9774	147.91	13683	120.53	16754	110.48

`Moment Curve`

Input size	$d=2$ cr #	$d=2$ time (s)	$d=4$ cr #	$d=4$ time (s)	$d=6$ cr #	$d=6$ time (s)	$d=8$ cr #	$d=8$ time (s)	$d=10$ cr #	$d=10$ time (s)
10000	57	58.51	324	11.68	807	7.9	1028	6.47	1354	6.12
25000	89	275.96	706	37.35	1698	24.08	2642	22.79	3411	20.62
50000	132	916.39	1151	98.25	2608	61.06	4836	52.4	6263	44.79
100000	209	2978.21	2797	268.95	5502	161.1	7743	133.25	10713	113.01

Table 2 Summary of the experimental results for set systems induced by half-spaces on two input types.

Evaluation. We present our experimental results in Table 2. It shows the observed crossing numbers and running times on inputs of size up to 100000. We see that the algorithm becomes faster as the dimension increases (note that the crossing number increases with dimension). For example, in dimension 6, it takes only around 160 seconds to create a matching for 100000 points. We highlight again that this is the first implementation of matching construction in dimensions larger than 2, and that even in $\mathbb{R}^2$, this is a big step forward from previous experimental results that only considered inputs of size at most 159, see [16].

Implementation details. Recall that our algorithm maintains weights on each pre-sampled edge, and these weights can be halved at each iteration. Instead of storing these potentially exponentially small weights explicitly, we simply maintain a partition of the edges into groups such that each group consists of elements that have been updated the same number of times, and thus have the same weight. We store the (exponentially increasing) weights of the test set half-spaces in the same way. To sample an edge or a half-space with respect to the current weights, it suffices to sample from the heaviest $\Theta(\log n)$ groups. The remaining groups have $o\left(\frac{1}{n}\right)$-th fraction of the total weight, which can be shown to not effect the analysis. We perform an initial $\frac{n}{2}$ iterations to set more accurate edge weights and start constructing the final matching only afterwards.

Test set generation. Linear-sized test set that achieves the guarantee of Lemma 12 can be constructed via cuttings, which are impractical in higher dimensions. Since the study of test-sets is not the main focus of this work and to speed-up the computations, our implementation, builds the test set by $n \log n$ random d-tuples of the input points; Table 2 reports the crossing numbers with respect to this particular test set. We refer to [2] for a detailed overview on constructions and sizes of test-sets for various geometric objects.

6 Applications

Here we present applications from learning and graph theory.

Approximating sign rank. Let $(X, \mathcal{S})$ be a set system and let $A \in \mathbb{R}^{n \times m}$ be its signed membership matrix, that is, $(A)_{x,S} = 1$ if $x \in S$ and $(A)_{x,S} = -1$ otherwise. The *sign rank* of $(X, \mathcal{S})$ is defined as the minimum rank of a matrix having the same sign pattern as A. Geometrically, it captures the minimum dimension of a Euclidean space in which $(X, \mathcal{S})$ can be embedded and realized by half-spaces through the origin. This embedding is linked to the efficiency of many practical machine learning algorithms, such as support vector machines and kernel classifiers. Using a connection between the sign-rank and the crossing number of a spanning path established in Alon *et al.*[6], we get the following corollary.

▶ **Corollary 16.** *Let $(X, \mathcal{S})$ be a set system and let $a > 0, b$ and $\gamma \in \left[\frac{1}{\log n}, 1\right]$ such that any $Y \subseteq X$ has a spanning path with crossing number at most $a|Y|^\gamma + b$. Then there is a randomized algorithm that constructs an embedding of X into $\mathbb{R}^D$ with $D \leq \left(\frac{8a}{\gamma}\right) n^\gamma + 4b \log n$ in expectation such that each $S \in \mathcal{S}$ can be represented with a half-space in $\mathbb{R}^D$. The expected running time of the algorithm is upper-bounded by the time complexity of $O\left(n^{2-\gamma} \ln^2 n + mn^{1-\gamma} \ln m\right)$ calls to the membership Oracle of $(X, \mathcal{S})$.*

Approximating diameter of graphs. It is known that the diameter of a graph cannot be computed in subquadratic time under the Strong Exponential-Time Hypothesis [31]. However, the situation can be improved if we restrict ourselves to graphs with bounded VC-dimension. Given a graph $G = (V, E)$, its VC-dimension is defined as the VC-dim$(V, \mathcal{N})$, where $\mathcal{N} = \{N(v) \mid v \in V\}$ with $N(v) = \{u \in V \ : \ uv \in E\}$. Recently, Ducoffe *et al.*[13] proposed a subquadratic time algorithm for deciding whether a graph with bounded VC dimension has diameter 2. Their algorithm relies on constructing a spanning path of V with low crossing number with respect to $\mathcal{N}$ and has running time $\tilde{O}(|E| \cdot |V|^{1-\varepsilon_d})$, where $\varepsilon_d = (2^{d+1}[3(d+1) - 1] + 1)^{-1}$ and $d =$ VC-dim(G). Using our algorithm we can obtain the following mild improvement over their result.

▶ **Corollary 17.** *Let G be a graph with VC dimension bounded by a constant d. Then there is a randomized algorithm that decides whether G has diameter 2 in expected time* $\tilde{O}\left(|E| \cdot |V|^{1-1/2^{d+1}}\right)$.

References

1. P. K. Agarwal. Simplex range searching. In *Journey Through Discrete Mathematics*, pages 1–30. Springer, 2013.
2. P. K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete & Computational Geometry*, 11(4):393–418, 1994.
3. P. K. Agarwal, J. Matoušek, and M. Sharir. On range searching with semialgebraic sets. II. *SIAM Journal on Computing*, 42(6):2039–2062, 2013.
4. P. K. Agarwal and J. Pan. Near-linear algorithms for geometric hitting sets and set covers. In *Proceedings of Symposium on Computational Geometry*, SOCG'14, page 271–279, 2014.
5. N. Alon, D. Haussler, and E. Welzl. Partitioning and geometric embedding of range spaces of finite Vapnik-Chervonenkis dimension. In *SoCG '87*, 1987.
6. N. Alon, S. Moran, and A. Yehudayoff. Sign rank versus VC dimension. In *COLT*, 2016.
7. T. M. Chan. Optimal partition trees. *Discrete Comput. Geom.*, 47(4):661–690, 2012.
8. T. M. Chan, E. Grant, J. Könemann, and M. Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1576–1585.
9. B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, New York, NY, USA, 2000.
10. B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, page 467–489, 1989.
11. C. Chekuri and K. Quanrud. Randomized MWU for positive LPs. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, page 358–377, 2018.
12. C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding for matroid polytopes and applications. *arXiv preprint arXiv:0909.4348*, 2009.
13. G. Ducoffe, M. Habib, and L. Viennot. Diameter computation on h-minor free graphs and graphs of bounded (distance) VC-dimension. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1905–1922, 2020.
14. E. Ezra, S. Har-Peled, H. Kaplan, and M. Sharir. Decomposing arrangements of hyperplanes: VC-dimension, combinatorial dimension, and point location. *Discret. Comput. Geom.*, 64(1):109–173, 2020.
15. S. P. Fekete, M. E. Lübbecke, and H. Meijer. Minimizing the stabbing number of matchings, trees, and triangulations. In *Proceedings of Symposium on Discrete Algorithms (SODA)*, 2004.
16. P. Giannopoulos, M. Konzack, and W Mulzer. Low-crossing spanning trees: an alternative proof and experiments. In *Proceedings of EuroCG*, 2014.
17. M. D. Grigoriadis and L. G. Khachiyan. A sublinear-time randomized approximation algorithm for matrix games. *Operations Research Letters*, 18(2):53 – 58, 1995.
18. S. Har-Peled. Constructing planar cuttings in theory and practice. *SIAM J. Comput.*, 29:2016–2039, 2000.
19. S. Har-Peled. Approximating spanning trees with low crossing number. *arXiv*, abs/0907.1131, 2009.
20. S. Har-Peled. *Geometric Approximation Algorithms*. American Mathematical Society, Boston, MA, USA, 2011.
21. D. Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.
22. M. Matheny and J. M. Phillips. Practical low-dimensional halfspace range space sampling. In *Annual European Symposium on Algorithms (ESA)*, volume 112, pages 62:1–62:14, 2018.
23. J. Matoušek. Efficient partition trees. *Discrete & Computational Geometry*, 8(3):315–334, 1992.
24. J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Springer Berlin Heidelberg, 1999.
25. J. Matoušek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.

26 J. Matoušek, E. Welzl, and L. Wernisch. Discrepancy and approximations for bounded VC-dimension. *Combinatorica*, 13(4):455–466, 1993.

27 N. H. Mustafa. Computing optimal epsilon-nets is as easy as finding an unhit set. In *46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 87:1–87:12, 2019.

28 N. H. Mustafa, K. Dutta, and A. Ghosh. A simple proof of optimal epsilon-nets. *Combinatorica*, 38(5):1269–1277, 2018.

29 N. H. Mustafa and K. Varadarajan. Epsilon-approximations and Epsilon-nets. In J. E. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*. CRC Press LLC, 2017.

30 J. Pach and P. K. Agarwal. *Combinatorial Geometry*. John Wiley & Sons, New York, NY, 1995.

31 L. Roditty and V. Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. pages 515–524, 06 2013.

32 K. R. Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 641–648, 2010.

33 H. E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968.

34 E. Welzl. Partition trees for triangle counting and other range searching problems. In *Proceedings of Annual Symposium on Computational Geometry (SoCG)*, page 23–33, 1988.

35 E. Welzl. On spanning trees with low crossing numbers. In *Data Structures and Efficient Algorithms, Final Report on the DFG Special Joint Initiative*, page 233–249, 1992.